Density Based Navier Stokes Solver for Transonic Flows

Oliver Borm[1] Aleksandar Jemcov[2] Hans-Peter Kau[1]

[1]Institute for Flight Propulsion
Technische Universität München

[2]Independent Consultant

2011-06-14

Outline

Outline

Turbomachinery and compressible transonic flows

- Implementation of a Godunov-like solver
- Approximate Riemann Solver:
  - Roe & Pike scheme with Harten's entropy fix [Roe81]
  - Roe & Pike ALE formulation [DCC98], [GATA06]
  - HLLC formulation from Batten et. al [BLG97]
  - HLLC ALE formulation from Luo et. al [LBL04]
  - AUSM+ ALE formulation from Luo et. al [LBL04]
  - Rusanov (Local Lax-Friedrichs) flux formulation
- 2nd Order space accuracy
- Local and Dual Time Stepping
- Multi-Stage Runge-Kutta Time Stepping
- Adapted total boundary conditions for internal flows
- Extension of MRF and SRF models
- Extension to CHT and FSI multiphysics solver

Riemann solvers

- Input - primitive variables
- Output - conservative fluxes (computed internally from primitive variables)
- Boundary Condition formulated for primitive state vector → Riemann solver is fed with this state vector to compute conservative fluxes at boundary faces
- GGI is working, as Riemann solver uses primitive variables as input
- Turbulence Modelling: Added as diffusive fluxes

Outline

Favre Averaged Navier Stokes Equation for Rotating Frames

$$\frac{\partial' \varrho}{\partial t} + \nabla \bullet (\varrho \vec{U}_{rel}) = 0$$

$$\frac{\partial' \varrho \vec{U}}{\partial t} + \nabla \bullet (\varrho \vec{U}_{rel} \otimes \vec{U}) + \nabla p = -\varrho \left(\vec{\omega} \times \vec{U} \right) + \nabla \bullet \sigma$$

$$\frac{\partial' \varrho E}{\partial t} + \nabla \bullet ((\varrho E + p) \vec{U}_{rel} + p \vec{U}_{rot}) = \nabla \bullet \left(\sigma \bullet \vec{U} \right) - \nabla \bullet \vec{q} + \nabla \bullet (\mu + \beta^* \mu_T) \nabla K$$

with $\vec{U} = \vec{U}_{rel} + \vec{U}_{rot}$ and $\vec{U}_{rot} = \left(\vec{\omega} \times \vec{x'} \right)$

Total Energy: $E = c_v T + 0.5 |\vec{U}|^2 + K$

Fourier heat law: $\vec{q} = -\lambda \nabla T$

Shear Stress Tensor (Newtonian Fluid):

$$\sigma = (\mu + \mu_T) \left(\nabla \vec{U} + \left(\nabla \vec{U} \right)^T - \tfrac{2}{3} (\nabla \bullet \vec{U}) \mathbb{G} \right) - \tfrac{2}{3} \varrho K \mathbb{G}$$

Basic Equations (II)

The LHS of this equation set are the Euler equations which can be formulated as ALE according to [LBL04] and expressed as

$$\frac{\partial}{\partial t}\int_V \vec{Q} dV + \int_A \vec{F} dA = RHS \tag{1}$$

with the conservative variables $\vec{Q}$ in each cell center and the convective numerical fluxes $\vec{F}$ at each face

$$\vec{Q} = \begin{pmatrix} \varrho \\ \varrho \vec{U} \\ \varrho E \end{pmatrix} \quad \vec{F} = \begin{pmatrix} \left[(\vec{U} - \dot{\vec{x}}) \bullet \vec{n}\right] \varrho \\ \left[(\vec{U} - \dot{\vec{x}}) \bullet \vec{n}\right] \varrho \vec{U} + p\vec{n} \\ \left[(\vec{U} - \dot{\vec{x}}) \bullet \vec{n}\right] (\varrho E + p) + p(\dot{\vec{x}} \bullet \vec{n}) \end{pmatrix}$$

with the velocity of each face $\dot{\vec{x}}$ which can be expressed as $\dot{\vec{x}} = \vec{U}_{rot}$ in the case of rotating machinery. Then the Geometric Conservation Law (GCL) is fullfilled automatically.

In case of non rotating faces, this scheme reduces to the well known Euler Equations.

Turbulence modeling for Rotating Reference Frames

It should ne noted that the symmetric part of the velocity gradients are identical

$$\nabla\vec{U} + \left(\nabla\vec{U}\right)^T = \nabla\vec{U}_{rel} + \left(\nabla\vec{U}_{rel}\right)^T \text{ and } \quad \nabla \bullet \vec{U} = \nabla \bullet \vec{U}_{rel} \qquad (2)$$

BUT the vorticities, skew symmetric part of the velocity gradients, are not identical:

$$\nabla\vec{U} - \left(\nabla\vec{U}\right)^T \neq \nabla\vec{U}_{rel} - \left(\nabla\vec{U}_{rel}\right)^T \qquad (3)$$

Care should be taken for turbulence models, which use the vorticity (skew(gradU)) (like in Spalart-Allmaras, realizableKE). As the result then depends on the chosen velocity for the turbulence model.
Using the relative velocity in the turbulence model is not a good choice in case of multiple blade rows. As the computation of the relative velocity gradient ($\nabla\vec{U}_{rel}$) at the rotor-stator interface (coupled patch) is error prune, as the relative velocity is jumping between two blade rows.

2nd order space accuracy - Slope Limiter

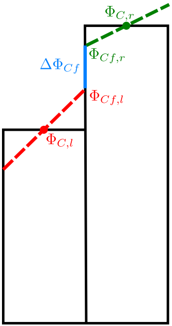
- Linear reconstruction of any input variable at faces as first term in Taylor series expansion from the cell centered value of this variable:

$$\Phi(x) = \Phi(a) + \frac{d\Phi(a)}{dx} \bullet [x - a] \qquad (4)$$

- Procedure is repeated for the left and right state vector of each face
- For stability and monotinicity reasons, the gradient has to be limited with a Limiter Ψ in the following way:

$$\Phi(Cf) = \Phi(C) + \Psi\{\nabla\Phi(C) \bullet [Cf - C]\} \qquad (5)$$

- multidimensional: Barth-Jespersen [BJ89], Venkatakrishnan [Ven95]
- one dimensional: Van Albada, Van Leer, Minmod



Slope Limiting

Runge-Kutta Time stepping

Eq. (1) can be in semi discrete form written with the residual vector $\vec{R}$

$$\frac{\partial \vec{Q}}{\partial t} + \vec{R}\left(\vec{Q}\right) = 0 \tag{6}$$

In order to obtain a steady state solution an explicit low storage multistage Runge-Kutta time stepping scheme [ALP93] is utilized, while n is the current physical time index

$$\vec{Q}^{(0)} = \vec{Q}^{n} \tag{7}$$

and advanced in multistage as with the multistage coefficent β_i

$$\vec{Q}^{(i+1)} = \vec{Q}^{(0)} + \beta_i \; \Delta t \; \vec{R}\left(\vec{Q}^{(i)}\right) \tag{8}$$

The next physical time step is the solution from last multistage

$$\vec{Q}^{(n+1)} = \vec{Q}^{(i_{max})} \tag{9}$$

Local Time stepping

In order to obtain a faster steady state solution a local time stepping, based on the CoEulerDdt scheme, was implemented. The maximum allowable compressible time step in each cell is computed by two parts, the inviscid and viscous time step

$$\Delta t_i = \frac{\Delta x}{|\vec{U}| + \sqrt{\frac{c_p}{c_v} RT}} \quad \text{and} \quad \Delta t_v = \frac{(\Delta x)^2 \varrho}{\mu + \mu_t} \tag{10}$$

The characteristic length scale can be computed with two different equations, either

$$\Delta x = \frac{V}{max(A_C)} \quad \text{or} \quad \Delta x = min \left| \left(\vec{Cf} - \vec{C} \right) \bullet \frac{\vec{Sf}}{magSf} \right|_{Cf} \tag{11}$$

The final time step is then computed like in [ALP93]

$$\Delta t = maxCo \frac{\Delta t_v \Delta t_i}{\Delta t_v + \Delta t_i} \tag{12}$$

Dual Time stepping

In case of unsteady simulations a dual (also referred to as pseudo) time stepping is used. Introduction of a new pseudo time into Eq. (1) leads to

$$\frac{\partial \vec{Q}}{\partial \tau} + \frac{\partial \vec{Q}}{\partial t} + \vec{R}\left(\vec{Q}\right) = 0 \tag{13}$$

In each inner iteration the solution is advanced in the pseudo time. If in Eq. (13) the additional pseudo time derivation vanishes, it becomes Eq. (1) . This approach leds to large physical time steps, which should speed up the time integration.
The first time step in the pseudo time integration is the last physical time step $Q^{j=0} = Q^n$. The first time step of the multistage integration inside the pseudo time is the last pseudo time $Q^{i=0} = Q^j$.

Dual Time stepping (II)

For a three point backward physical time integration and a multistage pseudo time integration, the solution is advanced in multistage pseudo time, with the multistage index i and the physical time index n

$$Q^{i+1}\left(\frac{1}{\beta_i \Delta\tau}+\frac{3}{2}\frac{1}{\Delta t}\right)=\left[\frac{1}{\beta_i \Delta\tau}Q^{i=0}+\frac{4}{2}\frac{1}{\Delta t}Q^{n}-\frac{1}{2}\frac{1}{\Delta t}Q^{n-1}\right]-R^{i} \tag{14}$$

The next pseudo time step is equal to the last multistage $Q^{j+1}=Q^{i_{max}}$.
Furthermore the next physical time step is equal to the last pseudo time step $Q^{n+1}=Q^{j_{max}}$.
The solution is only advanced in the multistage pseudo time integration.
The physical time steps are treated as explicit source terms.

Top Level Solvers

transonicMRFDyMFoam / transonicMRFCHTDyMFoam

- runs parallel, also with moving mesh
- steady and unsteady (multirows)
- CHT: heat transfer on non-conformal patches

transonicSteadySRFFoam

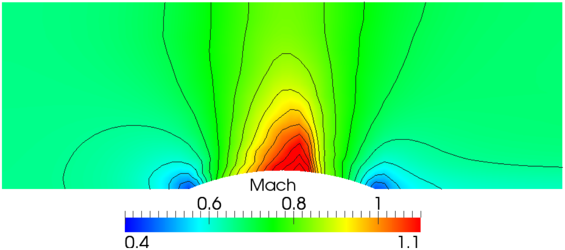
- runs parallel
- steady (single row)

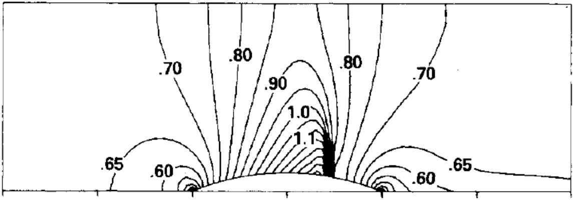
transonicMRFFSIDyMFoam (experimental)

- runs only serial, mainly due to mesh motion problems
- steady, mainly due to mesh motion problems, but multirows
- restricted to mesquiteMotionSolver due to the need of cyclicGgi, overlapGgi and ggi BC's
- Problems of displacement interpolations from solid to fluid for large deformations
- solid: centrifugal force and thermal stresses included

Outline

Bump Testcase - Ni 1982





Original plot from [Ni82]

Transonic testcase from [Ni82]

HLLC ALE

local time stepping

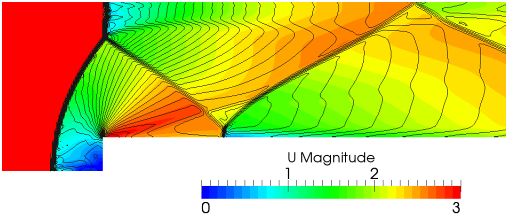
VK limiter with epsilon = 5

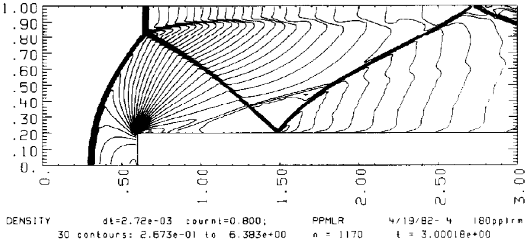
RKCoeff: 0.11 0.2766 0.5 1.0

maxCo = 1.5

meshes are identical

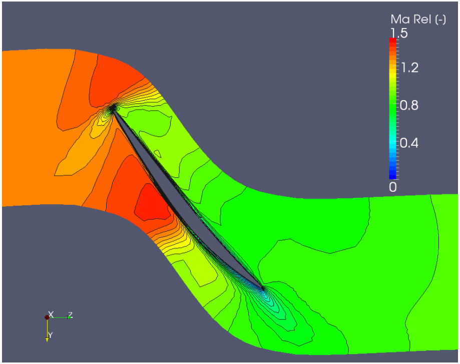
Forward Step Testcase - Woodward & Collela 1984



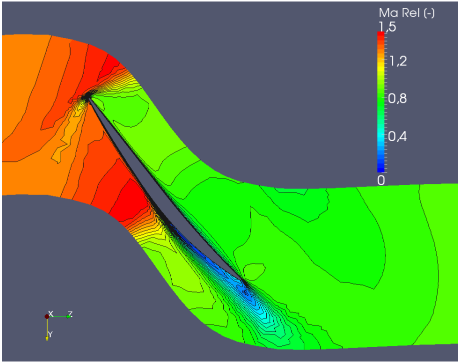


- Supersonic testcase and original plot from [WC84]
- magU with rho Isolines, in both cases isolines identical
- VK limiter with epsilon = 5
- meshes are identical
- dual time stepping with 50 inner iterations and local multistage time stepping
- results at t = 4s
- results depend highly on numerical setup (Limiter, Gradients, Riemann solver, etc.)

Testcase - NASA Rotor 37 SRF



transonicSteadySRFFoam

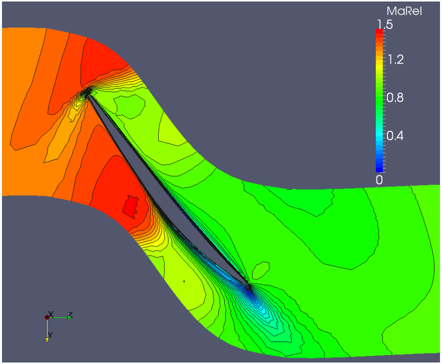


numeca

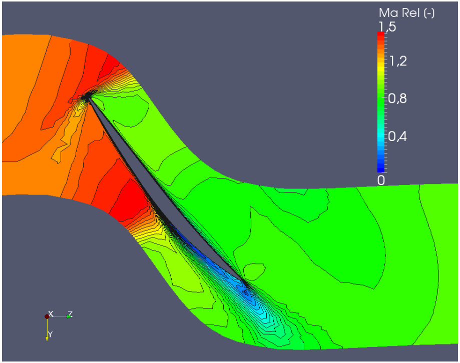
SRF HLLC (1st order), local time stepping, k-ω SST, same mesh

Numeca central JST scheme (2nd order), local time stepping, k-ω SST

Testcase - NASA Rotor 37 MRF



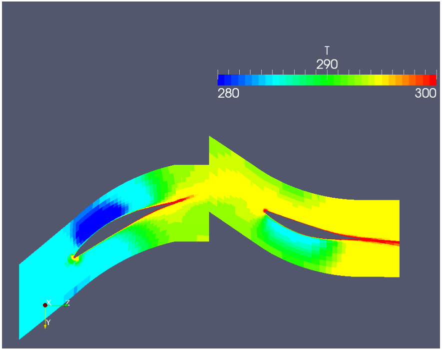
transonicMRFDyMFoam



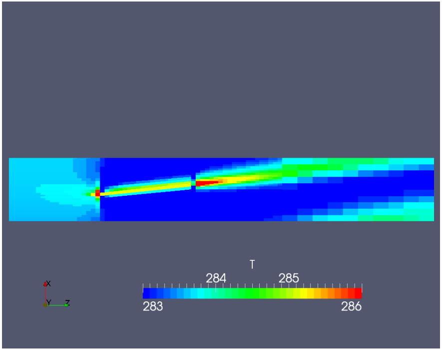
numeca

MRF HLLC ALE (2nd order) VK limiter, local time stepping, k-ω SST
Numeca central JST scheme (2nd order), local time stepping, k-ω SST

Unsteady MRF Testcases - axial compressor & CHT Mixer

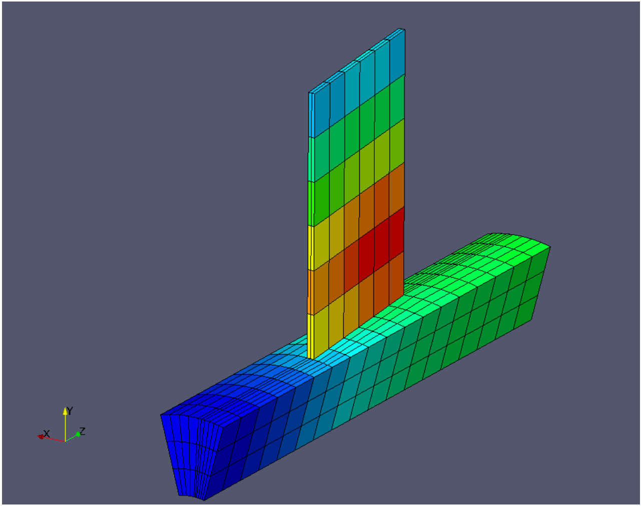


Unsteady MRF Axial Stage



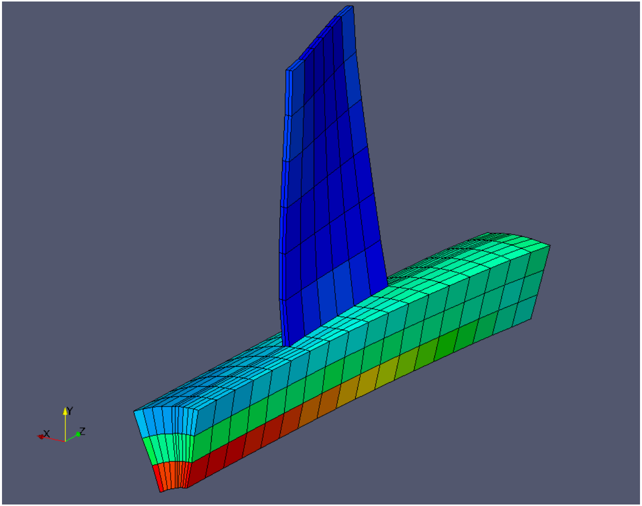
MRF CHT Unsteady Mixer

Generic Testcase - Steady Axial Mixer MRF CHT



Solid Temperature Distribution

Generic Testcase - Steady Axial Mixer MRF FSI



Displacement Scale Factor = 10
Equivilant Stress

Outline

Conclusion

Results depending highly on chosen parameters, especially for the 2nd order extension, and settings for them:

- Riemann solver
- Limiter
- Gradient calculation

What needs to be done:

- More validation cases
- RTSM for Riemann solver and multidimensional limiter
- Improve parallel mesh handling, especially for FSI solver
- Complete the Full-Approximation Storage (FAS) Multigrid
- Nice to have: Implicit time stepping

```
git clone git://openfoam-extend.git.sourceforge.net/
gitroot/openfoam-extend/DensityBasedTurbo
```

Outline

Literature I

Andrea Arnone, Meng-Sing Liou, and Louis A. Povinelli.
Multigrid Time-Accurate Integration of Navier-Stokes Equations.
Technical Report NASA TM 106373, NASA, 1993.

Timothy J. Barth and Dennis C. Jespersen.
The Design and Application of Upwind Schemes on Unstructured Meshes.
In *27th Aerospace Sciences Meeting*, number AIAA-89-0366, 1989.

P. Batten, M. A. Leschziner, and U. C. Goldberg.
Average-State Jacobians and Implicit Methods for Compressible Viscous and Turbulent Flows.
Journal of Computational Physics, 137:38–78, 1997.

Denis Darracq, Steeve Champagneux, and Alexandra Corjon.
Computation of Unsteady Turbulent Airfoil Flows with an Aeroelastic AUSM+ Implicit Solver.
In *16th Applied Aerodynamics Conference*, number AIAA-98-2411, 1998.

Gönç, Ak, Tuncer, and Aksel.
Computation of Turbulent Flows around Rotating Bodies Using Unstructured Grids.
In *24th Applied Aerodynamics Conference*, number AIAA-2006-2825, 2006.

Hong Luo, Joseph D. Baum, and Rainald Löhner.
On the computation of multi-material flows using ALE formulation.
Journal of Computational Physics, 194:304–328, 2004.

Literature III

Ron-Ho Ni.
A Multiple-Grid Scheme for Solving the Euler Equations.
AIAA Journal, 20(11):1565–1571, November 1982.

P. L. Roe.
Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes.
Journal of Computational Physics, 43:357–372, 1981.

V. Venkatakrishnan.
Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters.
Journal of Computational Physics, 118:120–130, 1995.

Literature IV

Paul Woodward and Phillip Colella.
The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks.
Journal of Computational Physics, 54:115–173, 1984.